AN EXPLORATORY ANALYSIS OF TIME SERIES ECONOMETRIC DATA FOR RETENTION FORECASTING USING DEEP LEARNING

THESIS

John C. O'Donnell, Second Lieutenant, USAF

AFIT-ENS-MS-22-M-159

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENS-MS-22-M-159

AN EXPLORATORY ANALYSIS OF TIME SERIES ECONOMETRIC DATA FOR RETENTION FORECASTING USING DEEP LEARNING

THESIS

Presented to the Faculty
Department of Operations Research
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Operations Research

John C. O'Donnell, B.S. Mathematics
Second Lieutenant, USAF

March 2022

AFIT-ENS-MS-22-M-159

AN EXPLORATORY ANALYSIS OF TIME SERIES ECONOMETRIC DATA FOR RETENTION FORECASTING USING DEEP LEARNING

THESIS

John C. O'Donnell, B.S. Mathematics
Second Lieutenant, USAF

Committee Membership:

Dr. Raymond Hill, Ph.D.
Chair
Dr. Bruce Cox, Ph.D.
Member

AFIT-ENS-MS-22-M-159

Abstract

Officer retention in the Air Force has been researched many times in an attempt to better predict the personnel needs of the Air Force for the future. There has been previous work done in regards to specific AFSCs and how their retention compares to specific yet similar private sector jobs. This study considers different econometric time series statistics as a feature space and an average Air Force officer separation rate as the response variable for the multivariate time series analysis deep learning techniques. The econometric indicators used in this study are New Business Formations, and the Consumer Confidence Index. The techniques considered for this study were Long Term Short Memory(LSTMs) models and Gated Recurrent Unit(GRU) models. This study shows that both GRUs and LSTMs perform fairly well with a forecast of 14 months out, but does not perform well comparatively to the more traditional univariate time series forecasting techniques, ARIMA models. The career fields with better performing models were career fields that will have jobs outside of the Air Force that will be more likely to hire in a period of economic growth, which would in turn increase the separation rate.

AFIT-ENS-MS-22-M-159

This research is dedicated to my family, friends, and especially my wife and puppies for being the support system that I needed to complete this endeavor.

Acknowledgements

This work would not have been possible without the constant support, guidance, and assistance of my advisor Dr. Raymond Hill throughout this research process. I would also like to thank my classmates and friends for making this program a much more enjoyable and feasible experience.

John C. O'Donnell

Contents

Page

List of Figures

Figure Page

List of Tables

AN EXPLORATORY ANALYSIS OF TIME SERIES ECONOMETRIC DATA FOR RETENTION FORECASTING USING DEEP LEARNING

I. Introduction

1.1 Background

The Air Force has experienced longstanding retention issues. Too often, talented individuals separate from the Air Force after their Active Duty Service Commitment (ADSC) has been completed. In a healthy economy, there are plenty of private sector jobs for military members to pursue once they leave the Air Force. While this may not cause a lack of personnel in the junior Company Grade Officer (CGO) positions, it does cause a gap in the time between the average ADSC for new personnel and the personnel that have passed the 10 year mark of their career, or are more than halfway to retirement. In an unhealthy economy there are far fewer civilian jobs, resulting in a higher retention rate. Programs such as PALACE CHASE help reduce the number of people in the Air Force when too many retain. The PALACE CHASE program allows current Active Duty Service Members to exchange on a one-to-one ratio of remaining Active Duty Service Commitment for an Air Reserve Component position[1]. This program helps to preserve the integrity of the mission, while still reducing the number of people in the Air Force to congressionally mandated levels.

The economy is difficult to predict. The National Bureau of Economic Research [2] is a private, non-partisan research organization, and uses the business cycle as a measure of the state of the economy. The National Bureau's Business Cycle Dating Committee maintains a chronology of U.S. Business cycles. The most recent committee announcement in June 2020 identified the economic peak of February 2020 before the sharp decline in March 2020 of the economic expansion that began after the Mortgage Backed Securities Housing Crisis in June of 2009 as the most recent “trough”. A peak is defined as an economic relative maximum point ,and a trough is an economic relative minimum point. Forecasting peaks and troughs are two separate trains of thought. If we consider the peak in 2020, that was a product of continuous economic growth between new businesses forming, interest rate cuts, and a relative minimum unemployment rate among other factors. A trough is more difficult to predict and might not necessarily be considered a likelihood. Take the Housing Crisis that was the cause of the trough in 2009 as an example. A key aspect of the crash was the misguided notion that people would always pay their mortgage. A more accurate statement is that well qualified buyers would pay their mortgage. The issue arose when unqualified applicants would be approved for mortgages that they would never have been able to afford. This problem would not have been evident to a majority of the public, and was unexpected to many people. Given the unexpected nature of this, it makes forecasting something like this even more difficult to predict.

Economic Analysis with statistical models use indicator variables. If the goal of the analysis is prediction then a leading economic variable is a useful variable in the model. This thought process does not necessarily lend itself to intuition. Unemployment rates are generally a good indicator of economic health, but given that unemployment rates measurable effects are only observable after they have occurred, it would not be useful as predictors so much as general analysis [3] .

A model with macroeconomic indicators was built by McGee [4] as an exploratory analysis to examine any significant relationships between leading economic variables and personnel data. The best model from her research included New Business Formations and New Durable Good Orders. These are both measures for the creation of new jobs for a healthy economy, which should be excellent predictors for the separation or retention of military personnel if those statistics are in the positive or negative direction.

Army retention has been modeled using time series forecasting techniques such as weighted averages, exponential smoothing, and auto-regression [5]. Time Series forecasting is a powerful modeling technique when predicting trends, and the challenges in previous studies have stemmed from not being able to discern what the underlying reasons were for the attrition of manpower. Without underlying reasons, policies cannot be adjusted accordingly.

1.2 Problem Statement

The retention problem for the Air Force has been examined using different techniques in the past such as survival analysis, regression, and Autoregressive Integrated Moving Average (ARIMA) models for selected individual AFSCs. This study utilized a multivariate time series forecasting approach using economic features to predict separation rate for the Air Force Officer Corps in the aggregate as well as selected AFSCs. Deep learning methods, specifically: Long Short Term Memory (LSTMs) networks and Gated Recurrent Units (GRUs) networks, have been successful in predicting nonlinear and noisy time series data such as econometric data. They may similarly be useful in finding the hidden relationships between the econometric data and the separation data. If we are able to use these models to predict increases in separation accurately enough, leadership could be able to implement changes to incentivize retention.

1.3 Document Overview

This document is organized as follows. Chapter II provides an overview of relevant background information, previous work that this study is expanding upon, and different models and techniques used. Chapter III provides the models and methodology for the problem. Chapter IV provides the results and comparison of the models. chapter V draws conclusions from the results in the previous chapters and provides future work recommendations.

II. Literature Review

There have been many studies examining the causality of Air Force retention issues. Some studies used economic indicators to model retention. Previous work to find significant predictors by using methods such as linear regression, logistic regression, or neural networks examined various indicators for models of future retention. McGee [4] found two significant economic factors that were useful in predicting future attrition. These are utilized in this study. Cross-referencing these time series models with the attrition rates over time should identify whether or not the predictive capabilities are significant.

This chapter will go over research on how the private sector handles attrition forecasting, relevant previous work done on Air Force Retention, time series forecasting approaches using deep learning and ARIMA, and background mathematics on Recurrent Neural Networks, Long Short Term Memory (LSTMS), and Gated Recurrent Units (GRUs).

2.1 Private Sector Employee Retention Forecasting

The private sector has an inherent advantage over the military in regards to the consequences of losing a higher ranking employee. If a company loses a manager or VP, they have the option of either promoting internally or hiring externally. The military only has the option to promote internally. The military also operates on an active duty service commitment meaning that separation from the air force has more of a time component than employee retention in the private sector. Given that, the forecasting research that has been done in the private sector has been based around classification models. These models use a combination of categorical and numerical data to predict if an employee is going to leave the company.

Pachal et al. [6] used an artificial neural network as a classification model to predict whether or not specific employees would have an intent to leave the company. This is a different approach from multivariate time series forecasting given that this is looking at multiple variables with a binary result rather than predicting a numerical value with respect to time. The majority of the features for the data in this study were categorical in nature. These features included characteristics such as, Gender, Marital Status, Number of Children, and Education. Some of the numerical features included number of years at the company, salary, and bonuses. The best neural network was able to correctly classify an employee's intentions 54.45% of the time. While this is better than a coin toss statistically, there is not enough of a real practical difference to say this model was successful.

Ben Yahia et al. [7] was another example of using machine learning and deep learning models to predict employee attrition. This study used an IBM simulated dataset, a kaggle dataset, and a real dataset from a questionnaire that was sent out to wide variety of demographics, but all worked at universities. These datasets included a combination of categorical data and numerical data that related to satisfaction of a current job as well as a binary classifier on if they had any intent to quit their current organization. They also had a balanced dataset in regards to the classification of intent to leave where 47.3% had an intent to leave their job and 52.7% had no intention of leaving. This is an important quality for the data to have given that an unbalanced dataset could cause the model to be more likely to predict the outcome that is more frequent in the data. As a whole the machine and deep learning models performed well, but the Random Forest performed the best with an accuracy of 0.98 before feature selection took place, and an accuracy of 0.983 after feature selection took place. This study also showed that the most important feature for prediction was how much an employee traveled for business. Another important feature was rewards and compensation, which are much more intuitive to predicting attrition - if someone does not feel like they're being compensated well enough, they will be more inclined to leave.

A classification model would be an interesting avenue to explore if sufficient data existed regarding an individual's demographics. It could be possible for the Air Force to send out surveys similar to what were used in these studies to supplement something like this in the future.

2.2 Relevant Previous Work on Air Force Retention

The retention problem in the Air Force sounds much simpler than it is. Intuition would suggest that the AFSCs with skills that would correlate well with the private sector would be more likely to separate since there would be a larger, possibly more lucrative job pool to choose from. However, if Air Force officers are not in AFSCs that directly correlate to the private sector, they are generally in management roles that themselves correlate to the private sector. The Air Force creates/attracts excellent leaders and skill-based workers alike that would most likely separate if given the option due to external factors such as familial and personal demographics or economic conditions

Gaupp [8] made an initial effort at using an agent-based model to examine economic indicators and pilot retention. Rather than making complicated specifications, Gaupp created a simpler model and,"let the model surprise them with the results." Guapp's results implied that the retention process for pilots is essentially a steady-state process over time, and is largely determined on a personal case-by-case basis. This led to the notion that instead of targeting individual characteristics and issues, the decision makers should focus more on the overall work environment.

Jantscher [9] created a baseline model for predicting retention rates for specific

AFSCs given economic metrics, and explored potential relationships between the Air Force retention rates and economic indicators. It was found that the majority of AFSCs had a negative correlation between economic indicators and their respective retention rates, meaning that as the economic indicators move in a positive direction, the retention rates move in a negative direction. The economic data that Jantscher utilized for this study corresponded to the ebb and flow of the economy; however, there was multicollinearity present, and due to the lack of economic indicator data points, a principal component analysis was not possible to attempt to rectify the multicollinearity issue. Jantscher found that the Intelligence (14N) and Chaplain (52R) career fields were positively correlated, which intuition says could be explained by the nature and opportunity of the work for that the military provides for intelligence officers, and the quality of life the military provides for Chaplains.

Elliot [10] focused on identifying statistically significant economic indicators affecting attrition and utilizing them to build a reliable mathematical forecasting model for attrition in the Air Force. Auto-Regressive Integrated Moving Average (ARIMA) models were utilized and confirmed previous results that lagging economic indicators can help predict retention. In particular, the unemployment rate, which was lagged by 24 months. Elliot's Dynamic Regression models performed better than the naive models at forecasting attrition, and was found statistically significant in all of the top performing models. Given the Air Force recommends making separation plans 12 months in advance, it would be necessary to implement a lagging indicator another 12 months in advance in order to get a more accurate prediction.

Schofield [11] examined sustainment behavior for each core AFSC. The Air Force generally follows the congressional mandates that are given for each career field that is deemed essential to operations. These sustainment lines are then optimized to allocate the right number of officers to each core AFSC by maximizing the number of officers in the lowest-manned career fields. Schofield notes three key assumptions in the current manning model, “The trends of historical retention, utilization, and crossflow of officers will continue for the next 30 years; The five year out congressionally mandated endstrength will not change for 25 years; and the five year out funded [manning] requirements will not change for 25 years”. [11] While Schofield notes that these assumptions are quite unrealistic, they are necessary for building the model.

Schofield [11] utilized survival functions to model retention. The survival models performed better with the Non-Rated Officer and Logistic Officer lines, and performed worse than the current model in the Support and Acquisition Career Fields. However, the dataset included 2014 which was considered a “force-shaping year”. Such data makes prediction a more complicated issue given the unnatural characteristics from that year. Schofield suggested Time Series Analysis or ARIMA models as a way to provide insight to attrition behavior over such a long span of time.

Pujats [12] expanded upon the Elliot’s work using AFSC attrition rates that would have correlating jobs in the private sector. He found that the attrition in the Air Force is not the same throughout all AFSCs, but should be considered individually. The significant factors for predicting this were generally civilian job prospects paired with general economic indicators such as GDP per Capita and Median Household Income. The 11X or Pilot, 17D or Cyber Officer, and 61A or Operations Research Analyst career fields did not correlate with external economic indicators given the current and growing need for those career fields regardless of the economic conditions. The 62E or Developmental Engineer career field was not found to be statistically significant with any particular career field given the variety of disciplines that engineering encompasses in the civilian sector. While the same could be said for Operations Research or 61A as well, they can be classified under a term that is defined by the Bureau of Labor Statistics that includes education, government, trade, financial activities, etc. as

"Employment in Services".

Pujats [12] asserts that forecasting human behavior is difficult given irrational thought processes and overall high level of unpredictability. He notes that prior to performing any forecasting based on economic factors, the individual officer should be studied to determine their likelihood of separation. However, Pujats also thought that performing individual evaluations would be a time consuming process and the issue of gathering enough quality data to train a model to perform such tests would be very difficult.

2.3 Artificial Neural Network Forecasting Studies

While Artificial Neural Networks are not the usual methodology for time series forecasting with deep learning, these studies show that deep learning architectures in general are able to outperform ARIMA models in regards to forecasting macroeconomic data.

2.3.1 Artificial Neural Network vs ARIMA Forecasting

Tang [13] compared the quantitative forecasting performance of Box-Jenkins vs Neural Networks. The three time series used in the study were a long memory pattern from Airline Passenger Data that shows an increasing trend and seasonal pattern, a somewhat unclear seasonal pattern time series for domestic car sales data, and an increasing trend time series with mostly irregular patterns for foreign car sales data. Tang concluded that neural networks were better in long-term forecasting, but Box-Jenkins was slightly better for short term forecasting. The hidden layer in the neural network was the difference maker between the two models because without it the neural network is functionally similar to the Box-Jenkins method. In summary the neural network would perform better with nonlinear data, but it would be more beneficial to use Box-Jenkins with a linear data set.

2.3.2 Artificial Neural Network Forecasting

A group of researchers from Singidunum University and Young Researchers and Elite Club from Islamic Azad Univeristy [14] conducted a study on how ANNs performed in the forecasting of economic parameters, those being Gross Domestic Product(GDP) and the Hirschman–Herfindahl Index (HHI). They used two different training techniques, Extreme Learning Machine(ELM) and Backwards Propagation(BP). An ANN with ELM was developed as a learning algorithm for a single hidden layer feed forward network. ELM and BP were applied to an ANN wth three layers, and when predicting GDP, based on the metrics of R^2 and Root Mean Square Error (RMSE), the ANN with the ELM training performed better than the BP trained ANN, but neither performed exceptionally well comparatively to the HHI prediction with R^2 values of 0.5609 for the ELM trained ANN and 0.443 for the BP trained ANN. The ELM and BP approach were used again and had an R^2 value of 0.9594 and 0.8402, respectively. The RMSE for both metrics for ELM was about half of the RMSE for BP, which shows that the ELM method of training shows promise for the prediction of macroeconomic indicators.

2.3.3 Hybrid ARIMA-Neural Network Approach

Zhang [15] proposed a hybrid approach to time series forecasting with ARIMA and Artifical Neural Network models. It is often difficult to determine whether or not a time series is generated from a linear or nonlinear underlying process, and real-world time series are rarely purely linear or nonlinear. In linear cases, ARIMA handles forecasting fairly well, but not as well in nonlinear cases, whereas ANNs handle nonlinear cases well, but have yielded mixed results in linear cases. As a generalization of the process, the first step would be to create an ARIMA model to analyze the linear portion of the problem and to create a neural network to model the residuals from the ARIMA model. Since the ARIMA model cannot capture the nonlinear structure of the data, the neural network is able to capture information about the nonlinearity from the residuals. This process was implemented on three data sets, the Wolf's sunspot data, the Canadian Lynx data, and the British Pound/USD exchange rate. All three data sets have different statistical characteristics a robust test of the hybrid model. After training and testing all three models, it was found that the hybrid model performed better than the component models in every scenario.

2.3.4 Recurrent Neural Networks

Reccurent Neural Networks or RNNs are a family of neural nets that are designed for sequential data such as time series data. It is structured similarly to a feedforward network, but unlike it, the input at time $t+1$ is the output of x_t and the input of x_{t+1}, which can be seen in Figure 1.

The RNN's output for a single instance can be shown mathematically as:

$$Y_{(t)} = \phi(W_x^T x_{(t)} + W_y^T y_{(t-1)} + b) \tag{1}$$

where $\phi()$ is the activation function, generally a "ReLu" function. "$W_x^T x_{(t)}$", "$W_y^T y_{(t-1)}$" are the weight matrices for the current time step input and the previous time steps, and lastly, b is the bias vector. The output can then be decomposed into matrix notation as:

$$Y_{(t)} = \phi([X_{(t)} Y_{(t-1)}]W + b) \tag{2}$$

where

$$W = \begin{bmatrix} W_x \\ W_y \end{bmatrix} \tag{3}$$

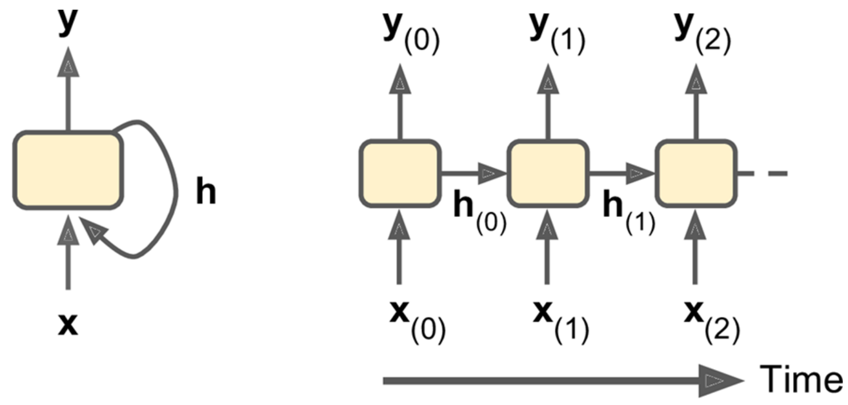


Figure 1: Simple RNN

2.3.4.1 Training RNNs

Training an RNN is less intutive than the Artificial Neural Network(ANN) training process. An ANN will make a foward pass through the network, determine error predictions, then backpropagate the error gradient back to the beginning of the network. RNNs work differently, using Back Propagation through time (BPTT). The BPTT process is depicted in Figure 2, where W is the weight, b is the bias, $Y_{(t)}$ is the output at a time step t, and $C(Y_{(t)'})$ is the cost function used to calculate the gradient. In Figure 2, the first two outputs are ignored, but every output that isn't greyed out will have the gradients of the cost function propagated backward through the unrolled network. The concept of an unrolled network is where every time step of a single cell of an RNN is shown. The model's weights are then updated based on the gradients across all time steps.

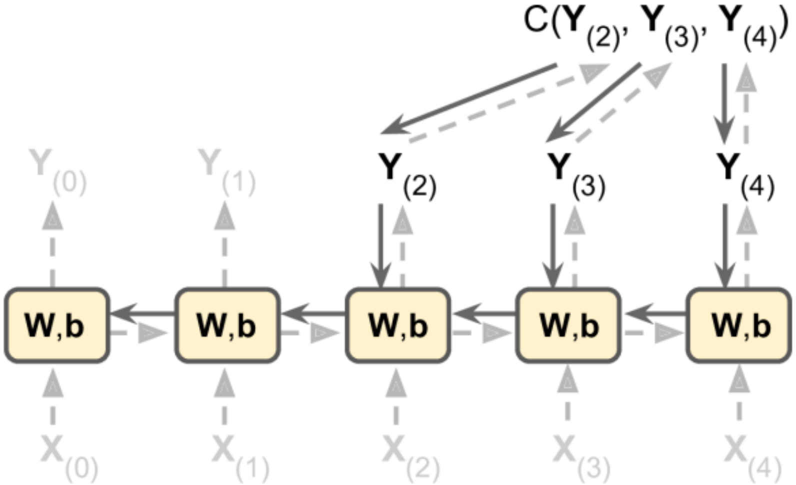


Figure 2: Backpropagation Through Time

2.3.4.2 Deep RNNs

Similarly to adding hidden layers to a feedforward network, deeper RNNs are used to abstract more relationships between the data. As can be seen in Figure 3, a deep RNN is essentially the stacking of recurrent layers on top of each other.

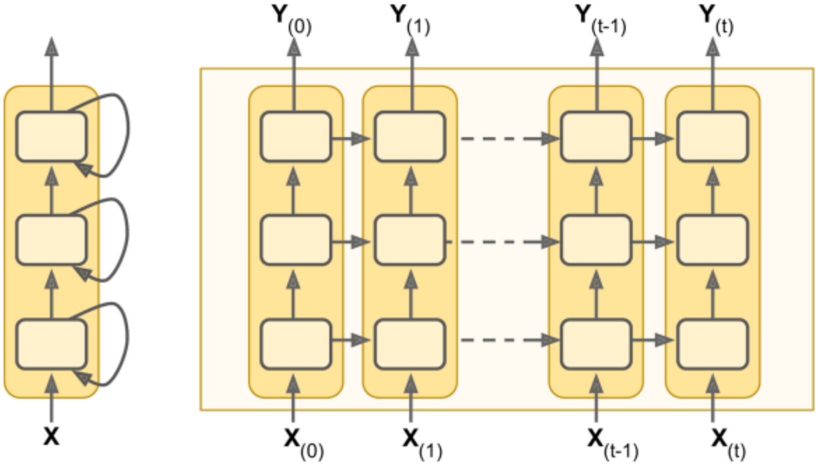


Figure 3: Deep RNN

A drawback of these Deep RNNs can be that the addition of too many stacked layers may hamper the model's performance. This problem is encountered when there are longer sequences of data. With longer sequences, the RNN will gradually forget the first inputs of the sequence

2.3.4.3 LSTMs and GRUs

Long Short Term Memory Models or LSTMs and Gated Recurrent Units or GRUs, are RNN-like structures used to address the vanishing gradient problem. The vanishing gradient is the problem that occurs when the backpropagation of the algorithm causes the gradient to get smaller and smaller until the weights no longer change and the gradient descent fails to converge.

An LSTM uses a simple multilayer perceptron (MLP) as the structure for an input gate, a forget gate, and an output gate. These define whether or not the data can pass through depending on priority and enable the network to partition data to save, forget, and remember. The structure of the LSTM is depicted in Figure 4.

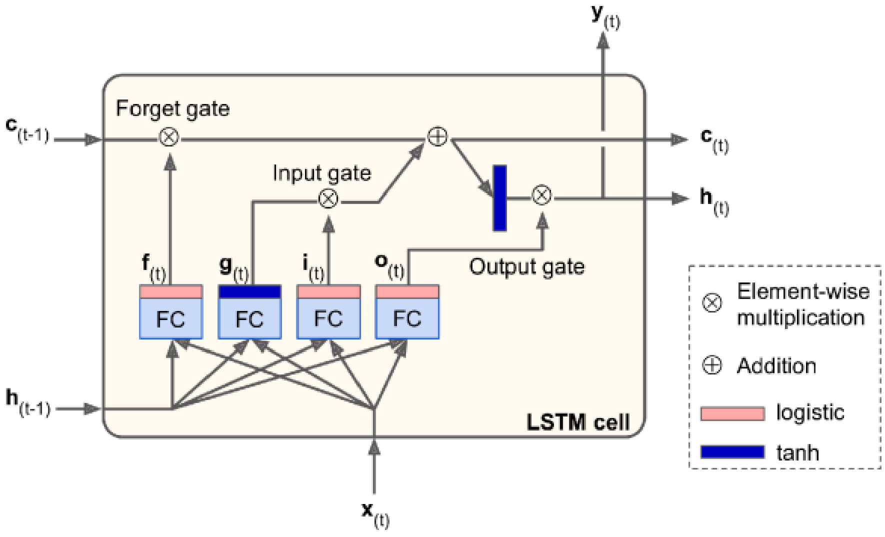


Figure 4: LSTM Network

The LSTM process can be explained mathematically as follows:

- The current timestep input X_t is combined with the previous time steps hidden state, $h_{(t-1)}$
- The forget gate, f_t, is equal to, $\sigma(X_t + h_{(t-1)})$ and relates to the long term memory
- The candidate cell state, $\bar{c}_t$ is equal to $tanh(X_t)$ and is multiplied with $\sigma(X_t)$ as the input gate, i_t
- The input gate controls what should be added to the long term memory
- The cell state for the current time step, c_t is then calculated by combining $(i_t \otimes \bar{c}_t + f_t \otimes c_{t-1})$
- The output gate, o_t, is also equal to $\sigma(X_t)$, and is then multiplied by $tanh(c_t)$, which outputs the hidden state for this time step, h_t

At each multiplication step in the process there is a weight and bias associated with each variable. As with RNNs, refining the weights through the training process improves the model. A summary of each equation follows.

$$\begin{aligned}
i_{(t)} &= \sigma(W_{xi}^T X_{(t)} + W_{hi}^T h_{(t-1)} + b_i) \\
f_{(t)} &= \sigma(W_{xf}^T X_{(t)} + W_{hf}^T h_{(t-1)} + b_f) \\
o_{(t)} &= \sigma(W_{xo}^T X_{(t)} + W_{ho}^T h_{(t-1)} + b_o) \\
\bar{c}_{(t)} &= \tanh(W_{x\bar{c}}^T X_{(t)} + W_{h\bar{c}}^T h_{(t-1)} + b_f) \\
c_{(t)} &= f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes \bar{c}_{(t)} \\
y_{(t)} &= h_{(t)} = o_{(t)} \otimes \tanh(c_{(t)})
\end{aligned} \tag{4}$$

The activation functions are used to simulate memory where values that would go to zero through the logistic or sigmoid are "forgotten" and conversely the inputs that go to one are "remembered". The hyperbolic tangent function is used to overcome the vanishing gradient given the necessity for a function to have a second derivative that can sustain a longer range before going to zero. This can be replaced by another function such as a "ReLu". The ReLu function logic is if the input is negative, the function returns zero, else it returns itself. The sigmoid function and hyperbolic tangent can be seen in equations 4 and 5 respectively.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{5}$$

$$tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{6}$$

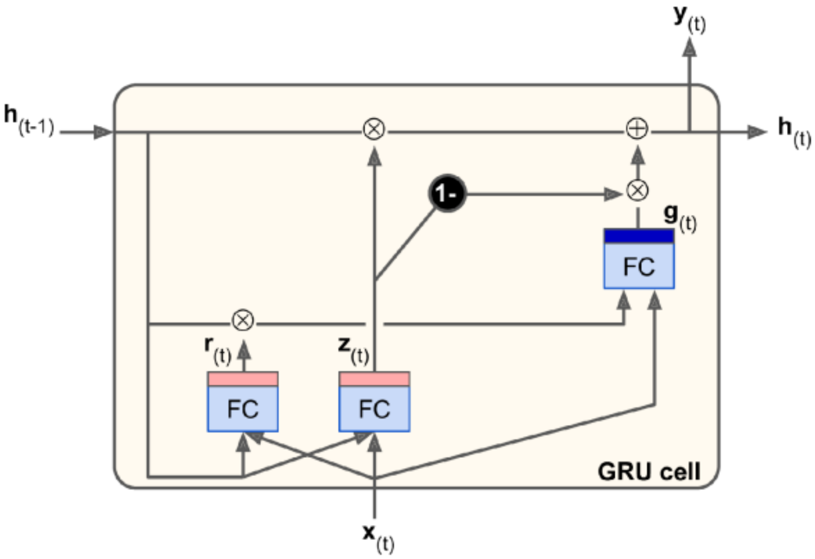


Figure 5: GRU Network

GRUs are a less complicated structure compared to an LSTM. Unlike an LSTM a GRU does not have an output gate, but has an update gate and a reset gate. These gates decide which information should be passed to the output. The structure can be seen in Figure 5.

The GRU process component equations are in equation 6.

$$\begin{aligned} r_{(t)} &= \sigma(W_{xr}^T X_{(t)} + W_{hr}^T h_{(t-1)} + b_r) \\ z_{(t)} &= \sigma(W_{xz}^T X_{(t)} + W_{hz}^T h_{(t-1)} + b_z) \\ g_{(t)} &= \tanh(W_{xc}^T X_{(t)} + W_{hc}^T h_{(t-1)} + b_c) \\ h_{(t)} &= z_{(t)} \otimes h_{(t-1)} + (1 - z_{(t)} \otimes g_{(t)} \end{aligned} \tag{7}$$

Unlike LSTMs, there is no output gate. There is a single gate controller, $z_{(t)}$, that controls the input gate and the forget gate. If it outputs a 1, the input gate is closed and the forget gate is opened. This is due to the $h_{(t)}$ in equation 6, where if $z_{(t)} = 0$, $h_{(t)}$ will equal $h_{(t-1)}$. [16]

2.3.5 Recurrent Neural Network Forecasting Studies

Tölö [17] used different Recurrent Neural Networks (RNNs) to predict financial crises one to five years ahead of time. Previously, literature used logistic regression and simple neural net architectures, but given RNNs' ability to make more robust predictions with Time Series data, RNNs can make significant improvements. The Jorda-Schularick-Taylor dataset was used in this analysis. Tölö compared the logit model and the Multilayer Perceptron (MLP) to a simple RNN, a Long Short Term Memory (LSTM) RNN, and a Gated Recurrent Units (GRU) network. A problem with a simple RNN is that they can be susceptible to vanishing or exploding gradients, so LSTMs and GRUs were introduced to avoid this problem by using gating mechanisms.

The measurement of Area Under the Curve (AUC) was used in Tölö's study, and the LSTM had the best overall performance across the different time horizons (1-5 years); however, there was a only a statistical difference between it and the MLP at the 5% significance level at the 2-year horizon. However, given a better accuracy overall for predicting what is an economic extremity, this method should be successful in predicting economic leading indicators and by extension officer retention curves.

Siami-Namini [18] compared the performance of ARIMA models vs LSTM models with different stock indexes such as the Dow Jones index, n225 index, Hang Seng Index, etc. The LSTM greatly reduced RMSE for the predictions. On average, The LSTM's prediction accuracy was 85% better than the ARIMA model's prediction accuracy.

Siami-Namini's study also analyzed the effect of the number of epochs on the RMSE for a prediction. An epoch is one full training run through of the data. There was no statistically significant difference on the financial data predictions when the number of epochs was changed, but that doesn't necessarily hold for all LSTM models given that some training data can be more difficult for the model to learn than others and can cause overfitting.

Di Persio and Honchar [19] used RNNs, LSTMs, and GRUs to forecast Google's daily stock price movement. They used data from the last five years to forecast up to a 5 day prediction horizon with up to 72% accuracy. The best prediction model was the LSTM with dropout added. Dropout is a method of regularization in a neural network model; it is used to prevent overfitting of the model. LSTMs and GRUs are both susceptible to overfitting of the training data due to the implementation of stacking to improve "memory".

III. Methodology

3.1 Methodology

This chapter discusses the data collection and preparation for both the econometric data as well as the AFSC separation data. The features are compiled into one data set where the econometric features are the explanatory variables and the separation rate data are the response variable. The multivariate forecasting techniques that are applied in this study are outlined with different data sets applied to each technique due to the differing computational requirements that accompany each one. The Deep learning models require much more time, resources, and data to properly train and tune their hyperparameters.

3.2 Multivariate Time Series Analysis

Multivariate methods are used to examine the relationship between the econometric data and the separation rate data. Gated Recurrent Units (GRUs) and Long Short-Term Memory models (LSTM) are the techniques chosen for this study, both of which are improved upon variants of Recurrent Neural Networks, and both methods use the econometric features as the independent feature space, and use the separation date as the dependent response.

3.3 Data Collection & Preparation

Data from HAF/A1 for the separation rate was aggregated by month, and then aggregated across the entire Air Force to create an Average Separation Rate for the entire Air Force Officer Corps. Given the limitations on data completeness, only 10 AFSCs were considered for the aggregate model and for individual models. This is seen in Table 1. This data covered 2005 until 2018 and contained 156 months of data points.

Table 1: All AFSCs considered in this study

AFSC List	
Air Force Specialty Code	**Title**
11X	Pilot
12X	Combat Systems Officer
21A	Aircraft Maintenance Officer
21R	Logistics Officer
32E	Civil Engineering
51J	JAG
62E	Developmental Engineer
63A	Acquisition Manager
64P	Contracting Officer
65F	Financial Manager

3.3.1 Data Transformations & Manipulations

The econometric data and the separation rate featured a variety of scales. All of the data were normalized to a $(0, 1)$ scale. The econometric features that were gathered from previous master data sets had six months of missing data, so an average imputation was performed to fill in the missing values. All of the data points were merged together by date. Given that the Separation Rate is aggregated as monthly data, the econometric data was also averaged into monthly data. An airman's decision to separate requires lead times, so the econometric data was lagged six months with respect to the separation data to provide a higher fidelity forecast. Other lag times were considered, but either worsened the autocorrelation problem with the data or yielded worse deep learning models.

3.4 Deep Learning Implementation

Deep learning has not been used to examine the Air Force Retention Problem. The method examined in this study was a Recurrent Neural Network, more specifically LSTMs and GRUs. Implementation involved the Tensorflow Keras library in Python with an EVGA XC3 Ultra Gaming RTX 3080 Graphics Processing Unit(GPU) and an Intel I5-10400 Central Processing Unit(CPU).

3.4.1 Neural Network Data Requirements

The deep learning model used an 80% training, 10% validation, and 10% test split, where the data are formatted as sequenced data without shuffling to keep the time series in order with the training data being the oldest data, the validation data following, and the test data are the most recent data. An additional requirement for the data are for it to be in a specific shape array. This requires using the "Numpy" or Numerical Python library in Python to create multidimensional Numpy arrays. The training data are in a multidimensional Numpy array of shape: (number of training points, window, number of features). The number of training points is the number of data points that are used to train the model. The "window" is a sliding window where the model will consider the past window of data points for the next prediction, meaning that given a point at time t, the window model will look back $t - window$ points to predict $t + 1$. The number of features is the total feature space including the independent and dependent variables.

3.4.2 Economic Feature Selection for LSTM

The standard for feature selection was the Durbin-Watson Statistic which is a serial correlation test for time series data. The baseline used was from McGee [4] who found statistical significance based on regression analysis for New Business Formations and New Durable Good orders. The Consumer Confidence Index(CCI) was added and returned the best Durbin-Watson value when all features were lagged by 6 months. Other economic statistics such as the unemployment rate, Housing Market Index, and US Treasury Yield rates were considered initially, but given the noisy nature of the data, they did not help the model's predictive power in any capacity.

3.4.3 Model Selection

A complex component of Neural Networks is the tuning of the hyperparameters. While most toy problems can be tuned manually, a larger problem like this with real world data, needs a more systematic approach to reach a broader range of hyperparameter combinations. The keras library makes sequential neural network models easy to use since it is possible to interchange LSTMs and GRUs by switching the names in the code. They also require the same shape parameters, so the comparison between the two models can be executed through a tuning process.

Neural Networks are commonly referred to as a "black box" with standard architectures that consist of an input layer, a type of layer - here LSTM or GRU layers, a dense layer with an activation function, an optimizer with a loss function, and a learning rate. This network also contains stacked LSTM layers which are defined in the command "return_sequences=True". These layers will allow each of the input neurons to have a hidden state output compared to the dense layer which will output the single value prediction for time $t + 1$. This network also has a dropout rate to help avoid the overfitting that LSTMs tend to have.

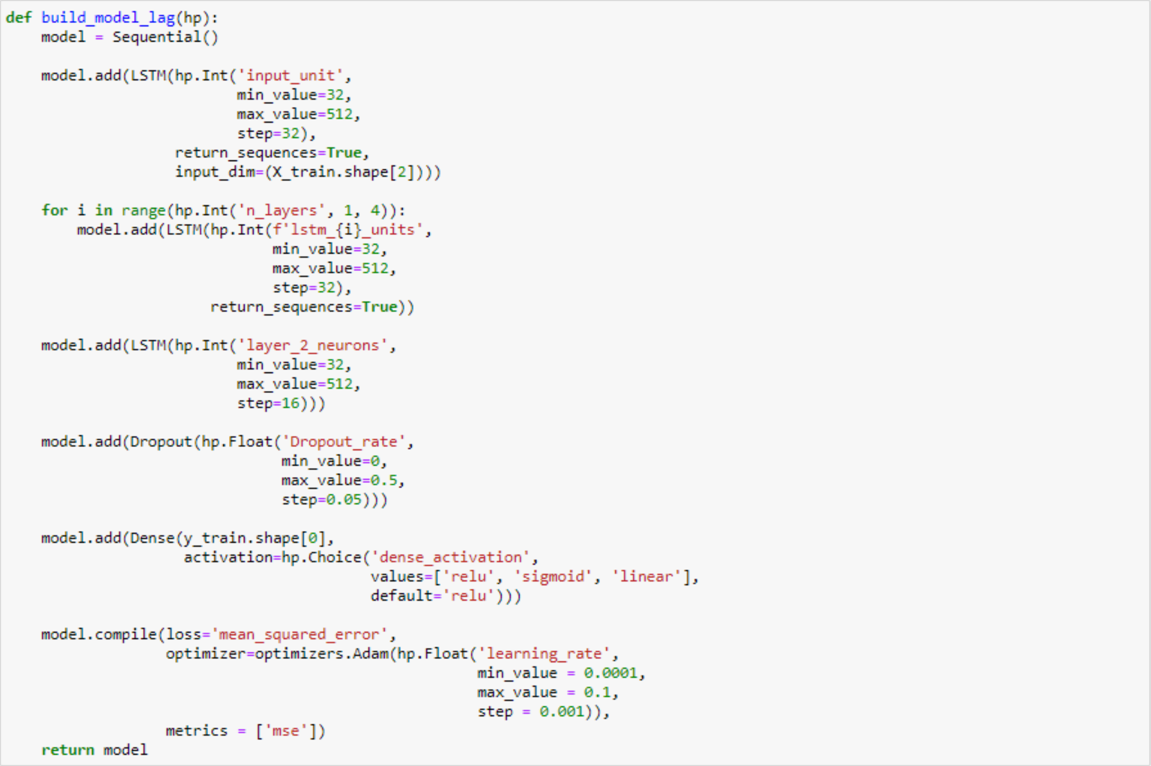
Each Neural Network is composed of several necessary components in their structure:

- An input layer in the correct shape for the specified type of Network, in this case LSTM or GRU layers
- A specified number of stacked layers;
- A method for regularization to help prevent models overfitting, such as dropout;
- A dense layer as well as a dense activation function to create an output for a time series to predict $t + 1$;
- An optimization algorithm, "Adam" was used to help with hyperparameter tuning since the default learning rate value of 0.001 can generally be used to success; and
- The loss function that the algorithm is used to train on, in this case was the mean squared error metric.

3.4.4 Keras Tuner Algorithm

The keras library uses the package "keras tuner" as an automated search algorithm. In this research, the hyperband, which is denoted as the current state of the art in the keras documentation, and the RandomSearch were used. The RandomSearch performed better than the hyperband search for the different deep learning models, so the RandomSearch algorithm is depicted in Figure 6. The search is tuning the number of neurons in the input layer, the number of stacked LSTM layers and the neurons in those layers, another LSTM layer's neurons, the dropout rate, the dense layer's activation function, and the learning rate. The "best" models for both LSTMs and GRUs were used as starting structures. This is due to the fact that this search could find a good model in terms of RMSE, but once that model is plotted it could just essentially return a trend line through the center of the data rather than an attempt to plot the separation rate peaks and troughs.

An example of the search function which is found in Figure 7, which is an example of the LSTM search. Every individual AFSC as well as the aggregated average separation separation rate went through 500 trials for 100 epochs for both an LSTM and GRU structure.



```
def build_model_lag(hp):
    model = Sequential()

    model.add(LSTM(hp.Int('input_unit',
                          min_value=32,
                          max_value=512,
                          step=32),
                   return_sequences=True,
                   input_dim=(X_train.shape[2])))

    for i in range(hp.Int('n_layers', 1, 4)):
        model.add(LSTM(hp.Int(f'lstm_{i}_units',
                              min_value=32,
                              max_value=512,
                              step=32),
                       return_sequences=True))

    model.add(LSTM(hp.Int('layer_2_neurons',
                          min_value=32,
                          max_value=512,
                          step=16)))

    model.add(Dropout(hp.Float('Dropout_rate',
                               min_value=0,
                               max_value=0.5,
                               step=0.05)))

    model.add(Dense(y_train.shape[0],
                    activation=hp.Choice('dense_activation',
                                         values=['relu', 'sigmoid', 'linear'],
                                         default='relu')))

    model.compile(loss='mean_squared_error',
                  optimizer=optimizers.Adam(hp.Float('learning_rate',
                                                     min_value = 0.0001,
                                                     max_value = 0.1,
                                                     step = 0.001)),
                  metrics = ['mse'])
    return model
```

Figure 6: The LSTM Search Function is used to find the optimal parameters of an LSTM model. This Search function can also be used to tune a GRU model if the names are switched at each instance of the code.

LSTMs and GRUs have been shown to perform better in multivariate time series forecasting than simple RNNs as was seen in Tölö [17] and Di Persio's [19] studies. So, they were not considered in this study due to the time consuming nature of tuning neural network hyperparameters.

3.5 ARIMA Model Baseline

The results of the deep learning techniques techniques were compared to each other as well as the a univariate forecast using an ARIMA model based solely on the separation rate data. This was used as a baseline because if a univariate model without econometric data performed better than a multivariate model with econometric then we may have reason to believe that the econometric data is not useful in predicting separation rate.

3.5.1 Stationarity and Differencing

Stationary data essentially means that the mean and variance stay the same over time, and is a necessary component for ARIMA models. Differencing of the data is the transformation usually used to obtain stationary data. Generally, the series will only need to be differenced a few times, which can be done as seen in equation 7. A first order differencing is denoted as $y\prime_t$ and a second order differencing is denoted as $y\prime\prime_t$.

$$y\prime_t = y_t - y_{t-1} \tag{8}$$

$$y\prime\prime_t = y\prime_t - y\prime_{t-1} = y_t - 2Y_{t-1} + y_{t-2}$$

LSTMs and GRUs don't necessarily need stationary data in order to learn or find the trends in a series due to their ability to find nonlinear relationships.

3.5.2 ARIMA Model Hyperparameters

ARIMA models are comprised of 7 components:

The Non-seasonal Components:

- p, or the “AR” in the ARIMA model, is the lag order of the model
- d, or the “I” in the ARIMA model, is the number of times the series is differenced to make it stationary
- q, or the “MA” in the ARIMA model, is the moving average order of the model

And the Seasonal Components:

- P, or the “AR” in the ARIMA model, is the lag order of the seasonal portion of the model
- D, or the “I” in the ARIMA model, is the number of times the seasonal portion of the model is differenced to make it stationary
- Q, or the “MA” in the ARIMA model, is the moving average order of the seasonal portion of the model
- m is the period for seasonal differencing, i.e. 4 for quarterly data, 12 for monthly data

3.5.3 Tuning ARIMA Hyperparameters

Tuning an ARIMA model’s Hyperparameters is done by examination of the Autocorrelation Function (ACF) plot and Partial Autocorrelation Function (PACF) plot. Candidate lag orders for p are found through the Autocorrelation Function (ACF) plot. As an example, consider Figure 7, an ACF plot for the 11X Career Field. The lags at 0, 1, 2, 3, and 12 are all considered statistically significant since they are outside of the blue shading which represents the significance level for different lag orders.

Tuning q is similar to p, but instead the PACF plot is examined. Consider the PACF plot for the same AFSC, 21A, in Figure 8. The points at 0, 1, 2, and 12

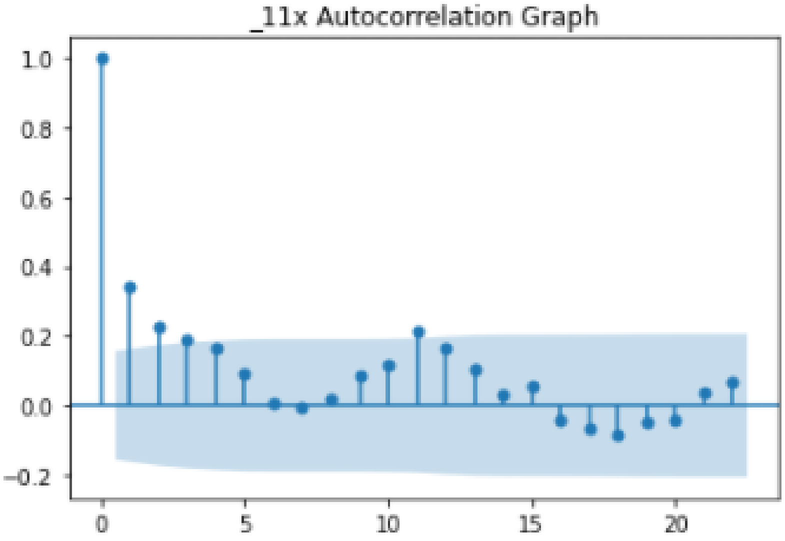


Figure 7: ACF Plot for 11X

are considered statistically significant in the model, but using higher order moving average orders can result in over fitting. So, only 0, 1, and 2 were considered.

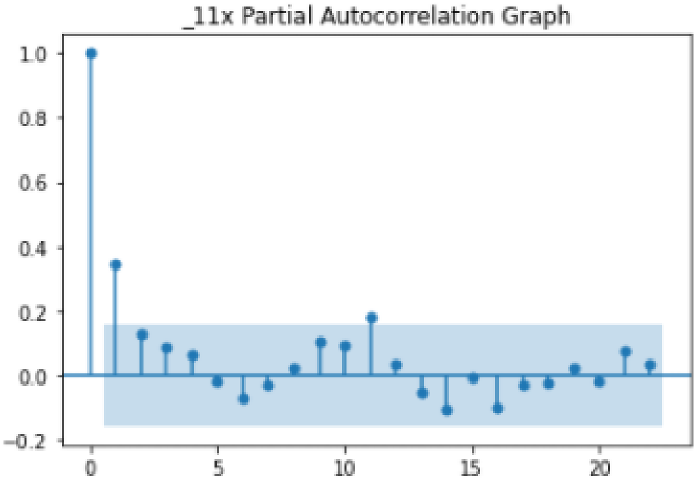


Figure 8: PACF Plot for 11X

Lastly, d is tuned by examining how many orders of differencing are required for the series to be stationary. This is done by using the Augmented Dickey-Fuller test, where if the p-value for every time series being considered is less than the significance level of 0.05, we can reject the null hypothesis that the series is not stationary.

Tuning the seasonal parameters for the models was done using the "auto_arima" function in python. This is an automated grid search algorithm that finds the best overall hyperparameters for an ARIMA model. These seasonal ARIMA models were compared to the non-seasonal ARIMA models to see if they were a better fit for the data.

3.5.4 Best ARIMA Model

The ARIMA model for each AFSC was chosen based on RMSE just like the LSTM and GRU models. The data was separated into the train and test split to where there were 14 months set aside for testing to compare this method to the multivariate methods.

IV. Results

4.1 Introduction

This chapter presents the analysis on the Deep Learning multivariate time series analysis methods, and how they compare to each other as well as the univariate time series models. The three different models used for comparison were ARIMA models for a baseline, GRU models, and LSTM models. The ARIMA model did not use econometric data since it is a univariate model. It used the separation rate as the data set where the future forecast is based solely on the past data. This is a useful baseline since the objective of this study is to determine whether or not econometric indicators are useful in forecasting separation rates. The initial comparison is for an aggregated dataset for all of the AFSCs in this study, and then a drill down into different AFSCs where one of the deep learning models performed better than the baseline ARIMA model.

4.2 Deep Learning Model Data Limitations

Training Neural Networks with a small amount of training data such as with this set will always present challenges. The main challenge with this data are validating the model with data that has no guarantee of containing similarities or traits that the training set has. The LSTM and GRU were both trained on 80% of the data with 10% set aside for validation, and 10% set aside for testing. There is also a sliding window of 10 data points that the model uses as the data that it is looking back on to predict the next point. This leaves the model with 124 data points for training, 14 for validation, 14 for testing. This small amount of data means that the model is not exposed to different specific trends over time. Regardless of hyperparameter tuning, the training and validation curves will most likely never converge. This would generally be seen as overfitting since the training loss is much smaller than the validation loss, but it will not be as much of a consideration moving forward due to the nature of the data.

4.3 RMSE Comparisons

The three different models were compared on the basis of RMSE. This measure is used to compare the time series models since differences in either direction can be potentially problematic for maintaining a sufficiently manned Air Force. As can be seen from table 2, the ARIMA models generally performed better than the deep learning models for most AFSCs as well as the aggregated data.

Table 2: RMSE Comparison between the two deep learning models and the baseline ARIMA model.

RMSE Comparison Between Models			
AFSC	**LSTM RMSE**	**GRU RMSE**	**ARIMA RMSE**
11X	.0023	**.0016**	.0021
12X	.0035	.0027	**.0019**
21A	.0045	.0044	**.0029**
21R	.0030	.0024	**.0021**
32E	**.0031**	.0041	.0032
51J	.0062	.0049	**.0041**
62E	.0025	.0025	**.0021**
63A	.0025	.0030	**.0021**
64P	.0039	.0033	**.0030**
65F	.0043	.0038	**.0031**
All AFSCs	.0030	.0021	**.0015**

4.4 Aggregated AFSC Data

While the GRU and LSTM models performed well, the ARIMA model performed better in terms of RMSE and much better at following the overall separation rate trend for the test data of the aggregated AFSC data as can be seen in Figures 11, 12, and 13. This aggregated data hides some of the noise that takes place in the individual

AFSCs. The problem with the deep learning models is that they are trying to model these hidden relationships, but they end up extrapolating noise and other error where its possible that none exists.

4.4.1 ARIMA Baseline

The candidate ARIMA model's were selected based on the ACF and PACF plots found in Figures 9 and 10, respectively. The significant points for the ACF plot that defines p show where p could be between (0-2, 11-12). The significant points for the PACF plot that defines q show where q could be between (0-1, 10-12). The higher values for p and q signal that there could be seasonality present in the data. Those seasonal spikes could also signal that the data is stationary. So, the Augmented Dickey-Fuller was conducted on the data and returned a value of 0.304. Given this, we would not have sufficient evidence to reject the null hypothesis that the data are non-stationary. Therefore, a non-seasonal ARIMA model was constructed with hyperparameters of (11,1,1), and returned an RMSE for the test prediction of 0.00152. To account for the possible seasonal components seen in the seasonal spikes of the ACF plot, a Seasonal ARIMA model was constructed with hyperparameters of (1,0,0)(1,0,1)[12] that returned an RMSE of 0.00252. Given a better performance on the test data, the non-seasonal ARIMA was chosen, and is shown in Figure 11.

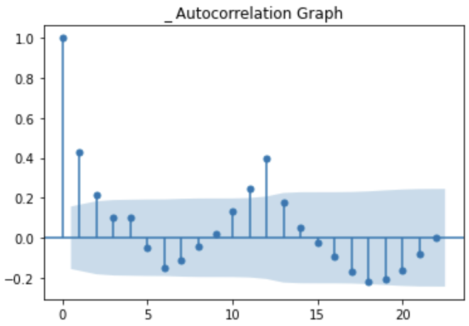


Figure 9: All AFSCs ACF Plot

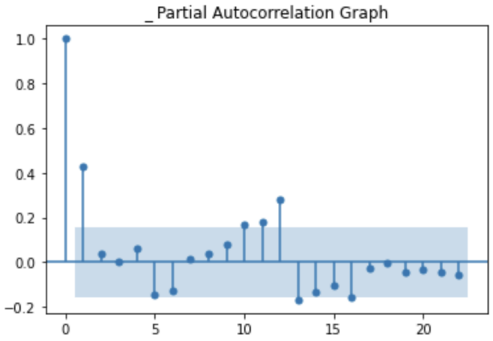
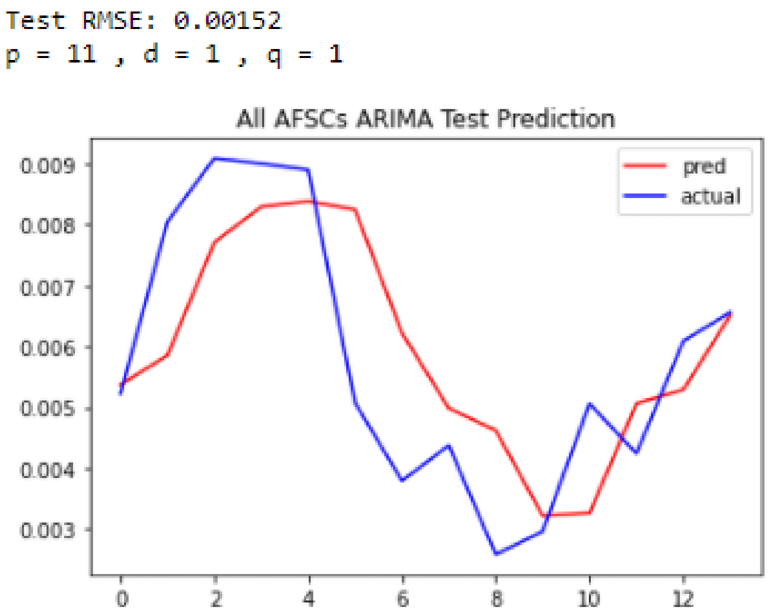


Figure 10: All AFSCs PACF Plot



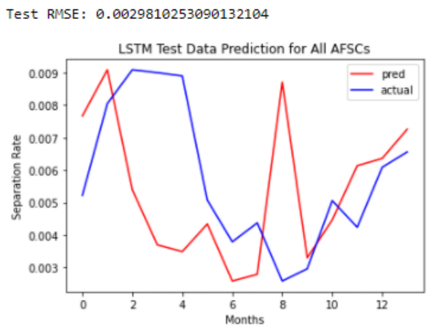
```
Test RMSE: 0.00152
p = 11 , d = 1 , q = 1
```

Figure 11: All AFSCs ARIMA Model

4.4.2 Deep Learning Models for All AFSCs

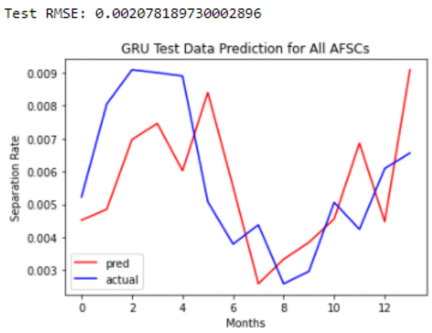
The Deep Learning models for the aggregated model were tuned initially for their optimized architectures as can be seen in figures 12 and 13. Unlike the ARIMA model, the hyperparameter tuning is more of a black box to where the best choices are more easily found through a grid search process - given sufficient computational power.

As can be seen in Figures 12 and 13, the GRU performed better than the LSTM in this problem instance. This can be confirmed by the test prediction RMSE. The test RMSE for the LSTM was 0.0030 whereas the test RMSE for the GRU model was 0.0021. However, neither model performed as well as the ARIMA model showing that the econometric features were not as strong of predictors as the separation rate itself.



```
Test RMSE: 0.0029810253090132104
```

Figure 12: All AFSCs LSTM Prediction



```
Test RMSE: 0.002078189730002896
```

Figure 13: All AFSCs GRU Prediction

4.5 LSTM vs GRU Models

Given how new the GRU structure of Recurrent Neural Networks is, the general consensus is that the LSTM will perform better on most data sets. However, this study found that the GRU performed better than the LSTM for 9 of the 11 data sets. This could be for a multitude of reasons. One possible answer is the stochastic nature of a neural network's training process. It is always possible for a network to reach a global minima, but given the small training set and randomness of the data, it is unlikely. Another possibility is that GRUs perform better than LSTMs with smaller amounts of training data. This is due to the GRU model using all of the data as its memory rather than an LSTM "forgetting" certain pieces of data if it does not match the criteria to get through the forget gate. This process for the LSTM is why longer time series sequences will generally perform better, but there is not a specific length where an LSTM model will always perform better than a GRU model. This makes it a necessity to sufficiently test both types of structures.

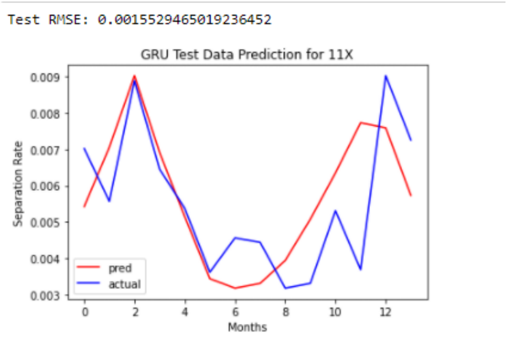
4.6 High Performing Deep Learning Models

While the majority of the deep learning models did not outperform the ARIMA models, there were two deep learning models that did perform better than ARIMA.

The GRU model for 11X and the LSTM model for 32E.

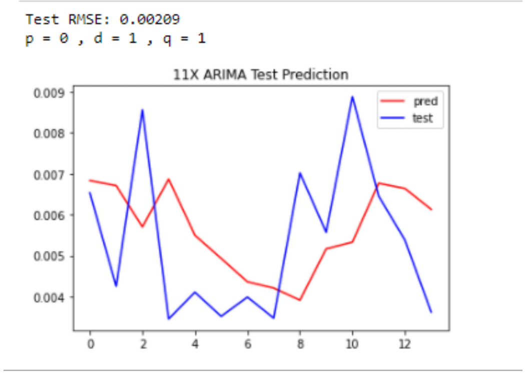
4.6.1 11X-Pilots LSTM Model

Pilot retention is always going to be a point of contention with the Air Force given the competition from the private sector for a more stable home life and possibly more competitive pay. The GRU model performing better than the ARIMA model here shows that the economic relationships the model was able to exploit were better predictors than the separation rate. On a more subjective level, as can be seen in Figures 14 and 15 the GRU model follows the overall trend much better than the ARIMA model does. The GRU model captured both separation peaks well enough to where a decision maker would be able to implement a change to either bring in enough new pilots from USAFA and ROTC or create more of an emphasis on pushing more through the training pipeline so that those students could fill the roles of the pilots that would be moving into staff roles.



```
Test RMSE: 0.0015529465019236452
```

Figure 14: 11X GRU Prediction

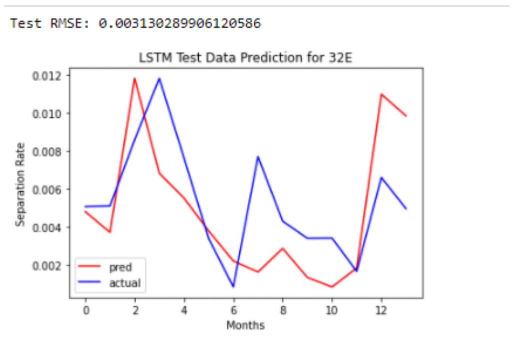


```
Test RMSE: 0.00209
p = 0 , d = 1 , q = 1
```

Figure 15: 11X ARIMA Prediction

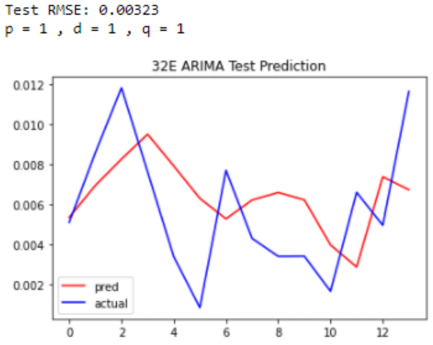
4.6.2 32E-Civil Engineering LSTM Model

Civil Engineering is a career field generally in demand in the private sector. Such a well defined private sector counterpart that could entice officers to separate in favor of a more stable civilian lifestyle and a possible pay increase. Unlike with the pilot data, the LSTM model performed much better than the GRU model and slightly better than the ARIMA model in terms of RMSE. In terms of following the trend, the ARIMA model does not capture the peaks of separation very well. It sticks to the middle of the data, staying between the minimum and maximum points. The LSTM does capture the first peak well, but struggles to capture the second and third peaks. While it does not capture them perfectly, if the second and third prediction peaks were averaged together, they would be about equal to the separation rate for that 5 month time period.



```
Test RMSE: 0.003130289906120586
```

Figure 16: 32E LSTM Prediction



```
Test RMSE: 0.00323
p = 1 , d = 1 , q = 1
```

Figure 17: 32E ARIMA Prediction

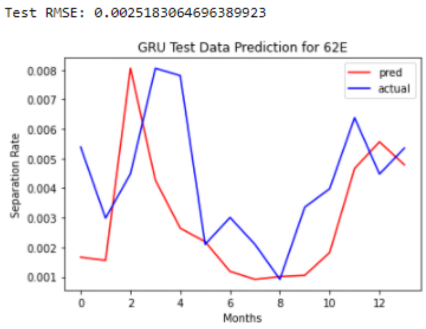
4.7 Potentially Useful Deep Learning Models

While these deep learning models did not perform as well as the ARIMA models in terms of RMSE, these models were able to follow the separation rate trend well enough to where they could be useful. The AFSCs that were selected for this section would also have a fairly stable private sector job if they were to separate from the Air Force.

4.7.1 62E-Developmental Engineers

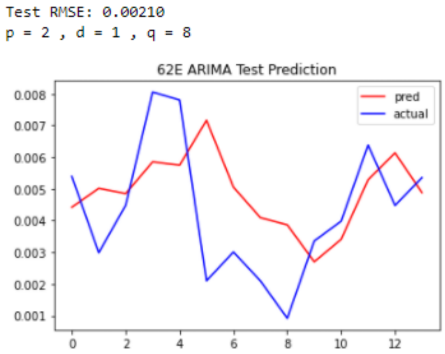
The developmental engineer is an AFSC that would most likely be able to find a job outside of the Air Force if they were to separate after completing their Active

Duty Service Commitment. The GRU model in Figure 17, was able to predict both peaks of attrition as well as the ARIMA model. This illustrates the point that while RMSE is a useful measure, it is calculated by differences at points in time. The average RMSE could be affected by the model predicting the peak one time step too early. In practice, this early peak prediction would not mean much since there would be a later adjustment to compensate for the increase in separations.



```
Test RMSE: 0.0025183064696389923
```

Figure 18: 62E GRU Prediction

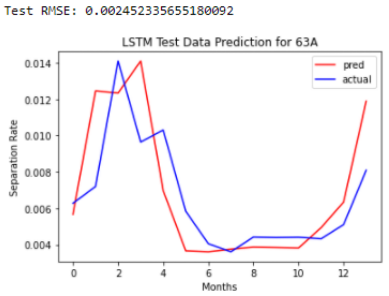


```
Test RMSE: 0.00210
p = 2 , d = 1 , q = 8
```

Figure 19: 62E ARIMA Prediction

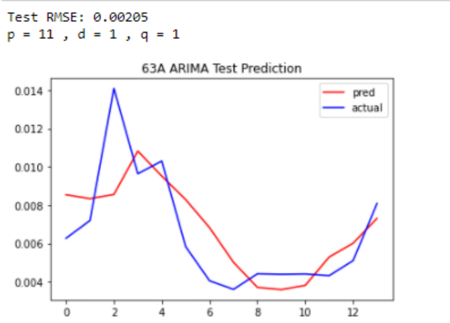
4.7.2 63A-Acquisitions Officer LSTM Model

As shown in Figures 20 and 21, the LSTM model follows the actual separation rate much better than the ARIMA model. Like the developmental engineer, an acquisitions officer will have similar job opportunities outside of the Air Force. Thus, as the economy enters a period of growth or recession, this could be an AFSC that needs to be monitored as the separation rate is sensitive to those economic changes.



```
Test RMSE: 0.002452335655180092
```

Figure 20: 63A LSTM Prediction

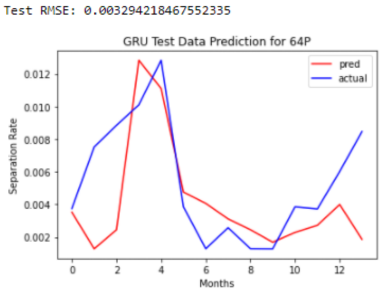


```
Test RMSE: 0.00205
p = 11 , d = 1 , q = 1
```

Figure 21: 63A ARIMA Prediction

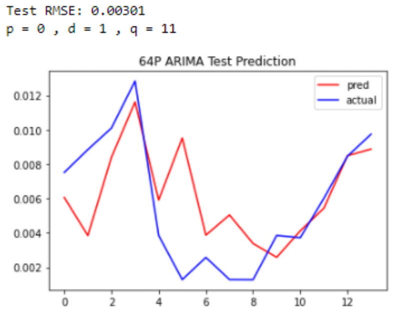
4.7.3 64P-Contracting Officer GRU Model

The GRU model for the Contracting officer was another example where the ARIMA model was able to capture a fairly steady increase in separation over time, but also predicted that there would be another sharp increase in separation when there would not be one. The GRU model was able to accurately capture the initial increase in separation, but was not able to forecast the secondary increase at month 12. This could lead to a possible shortage for the career field since the forecast would lead a decision maker to believe that an influx of contracting officers was not necessary.



```
Test RMSE: 0.003294218467552335
```

Figure 22: 64P GRU Prediction



```
Test RMSE: 0.00301
p = 0 , d = 1 , q = 11
```

Figure 23: 64P ARIMA Prediction

4.8 Omitted Graphs and Models

The training and validation loss plots for each deep learning model as well as the remaining prediction plots for the deep learning models and ARIMA models of the test data are found in the appendix. The loss function plots were removed given the differences between the training data and validation data caused the model to never converge by usual standards. A model will never converge if the validation data does not share enough similarities with the training data. The remaining models plots were omitted as they did not perform as well as the models above had.

V. Conclusions and Future Work

5.1 Conclusions

The goal of this research was to examine whether econometric data might help predict officer attrition. This was examined using multivariate time series analysis, specifically deep learning, to try and extrapolate the hidden relationships between the data that may not be possible to extrapolate with traditional forecasting methods, such as ARIMA models. Ten individual AFSC separation rates were considered in this study as well as an aggregated average separation rate for the AFSCs present in this study. The economic feature selection continued the work of McGee [4] and Elliott[10] by considering New Business Formations, New Durable Good Orders, and the Consumer Confidence Index lagged by 6 months and the unemployment rate lagged by 24 months. While the best Durbin-Watson scores were with models that contained the unemployment rate, the best deep learning models used only New Business Formations, New Durable Good Orders, and the Consumer Confidence Index.

While all possibilities for the hyperparameters could not be explored, every deep learning model was put through at least 500 trials of a random search algorithm for each AFSC. This gave a good starting point for the by-hand hyperparameter tuning process to be a fairly robust exploration of the best model. A difference between a "best" deep learning model and an ARIMA model is that the ARIMA model will be the same every instance given the same hyperparameters. A deep learning model will not be the same every instance given the stochastic optimization of the gradient. A different local minima could be found each time, and the global minima might never be found. The other trade off that has to be considered is how much easier the ARIMA model is to implement than the deep learning models. There are vastly reduced training times for the ARIMA models as well as far fewer hyperparameters.

One could very easily run an ARIMA model on a standard laptop, but to sufficiently run a deep learning model, you would need GPU or TPU performance due to the complexity of the matrix calculations that are taking place. Lastly, a limitation for these models is how little data are available for training. The 124 data points is not enough to capture all of the hidden relationships that these economic indicators could have with officer attrition. With all of this in mind, as well as the overall performance for the ARIMA model compared to the deep learning models, the ARIMA is a better choice for implementation for the time being until more data is available.

5.2 Future Work

Possible future work could be to explore data augmentation through the use of MIT's CTGAN library. The CTGAN library is a new library for data augmentation that generates synthetic tabular data meant to mimic real data with high fidelity. However, given the unpredictable nature of the attrition data, it may not be possible for the CTGAN to capture the underlying distribution of the data. In lieu of this, it could be possible to consider individual separation given specific demographics. If data exists that can give multiple instances of different demographics, the data augmentation process could be used to create multiples of these instances. That data could then be used to train a classification model to determine when that individual would separate.

Appendix A. Omitted AFSCs

This section contains the Training and Loss Curves, best deep learning model output, and best ARIMA model output.

A.1 12X-Combat Systems Officer

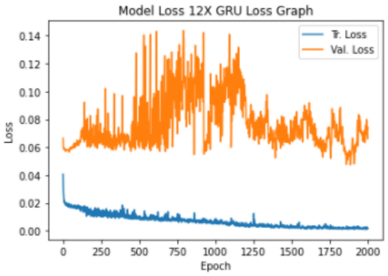
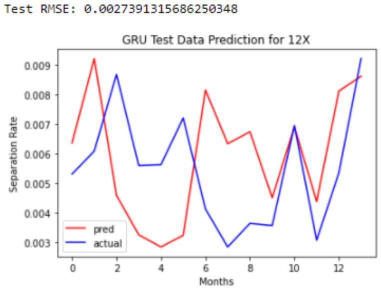
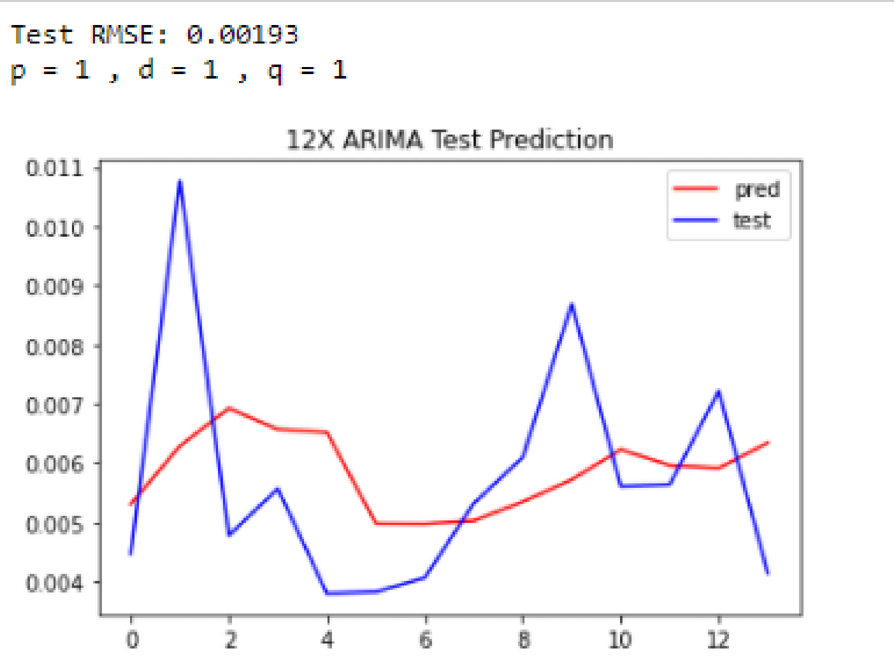


Figure 24: 12X GRU Training and Validation Loss Curves



```
Test RMSE: 0.0027391315686250348
```

Figure 25: 12X GRU Model Prediction and Test Plots



```
Test RMSE: 0.00193
p = 1 , d = 1 , q = 1
```

Figure 26: 12X ARIMA Model Output

A.2 21A-Aircraft Maintenance Officer

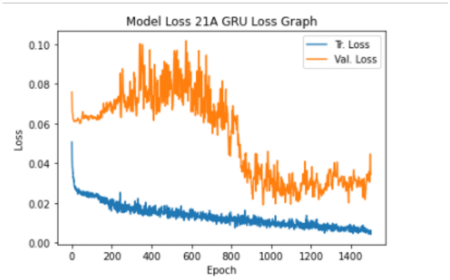
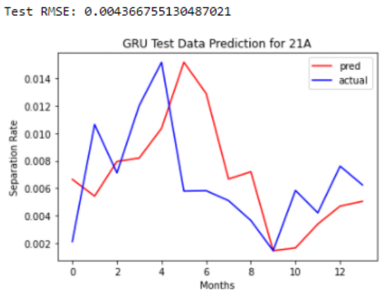
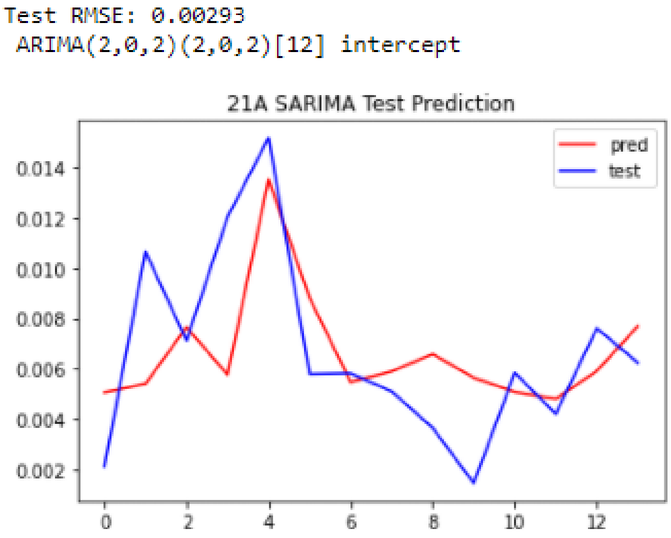


Figure 27: 21a GRU Training and Validation Loss Curves



```
Test RMSE: 0.004366755130487021
```

Figure 28: 21a GRU Model Prediction and Test Plots



```
Test RMSE: 0.00293
 ARIMA(2,0,2)(2,0,2)[12] intercept
```

Figure 29: 21A SARIMA Model Output

A.3 21R-Logistics Readiness Officer

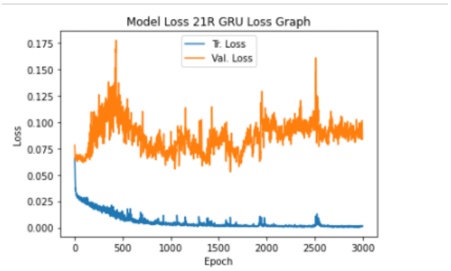
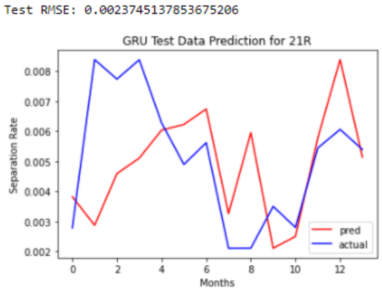
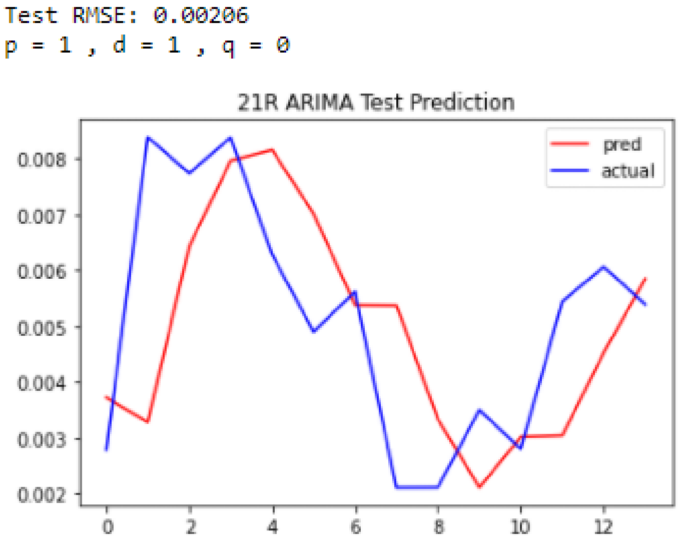


Figure 30: 21R GRU Training and Validation Loss Curves



```
Test RMSE: 0.0023745137853675206
```

Figure 31: 21R GRU Model Prediction and Test Plots



```
Test RMSE: 0.00206
p = 1 , d = 1 , q = 0
```

Figure 32: 21R ARIMA Model Output

A.4 51J-Judge Advocate

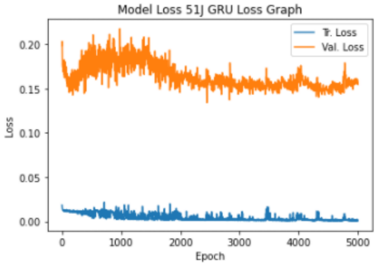
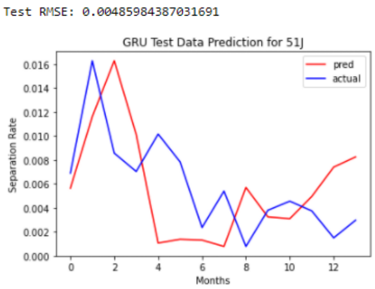
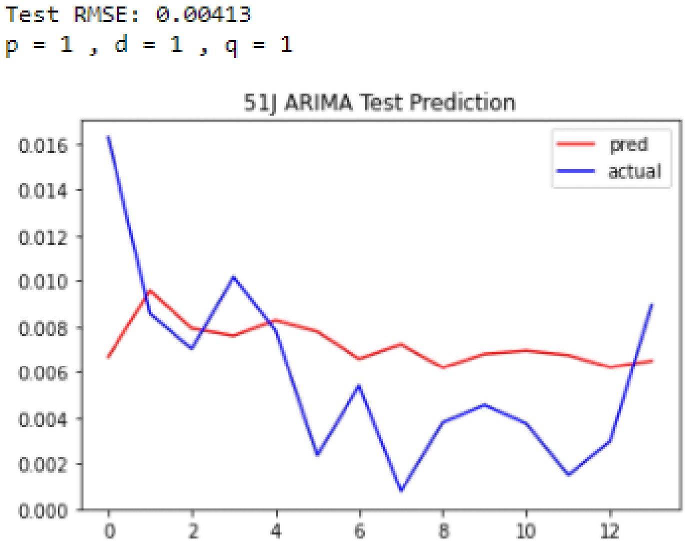


Figure 33: 51J GRU Training and Validation Loss Curves



```
Test RMSE: 0.00485984387031691
```

Figure 34: 51J GRU Model Prediction and Test Plots



```
Test RMSE: 0.00413
p = 1 , d = 1 , q = 1
```

Figure 35: 51J ARIMA Model Output

A.5 65F-Financial Readiness Officer

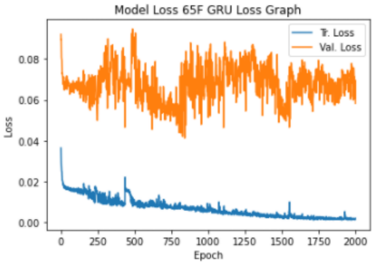


Figure 36: 65F GRU Training and Validation Loss Curves

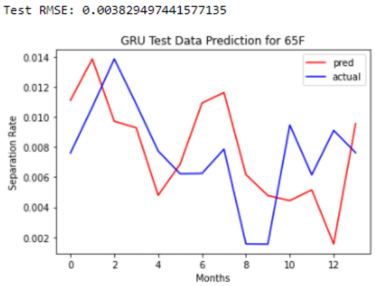

```
Test RMSE: 0.003829497441577135
```

Figure 37: 6F GRU Model Prediction and Test Plots

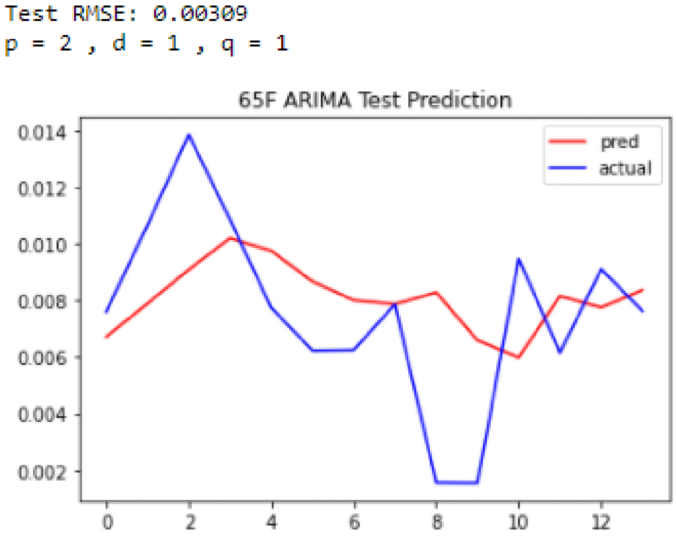

```
Test RMSE: 0.00309
p = 2 , d = 1 , q = 1
```

Figure 38: 65F ARIMA Model Output

Appendix B. Training and Validation Loss Graphs for Models Included in Thesis

B.1 11X GRU

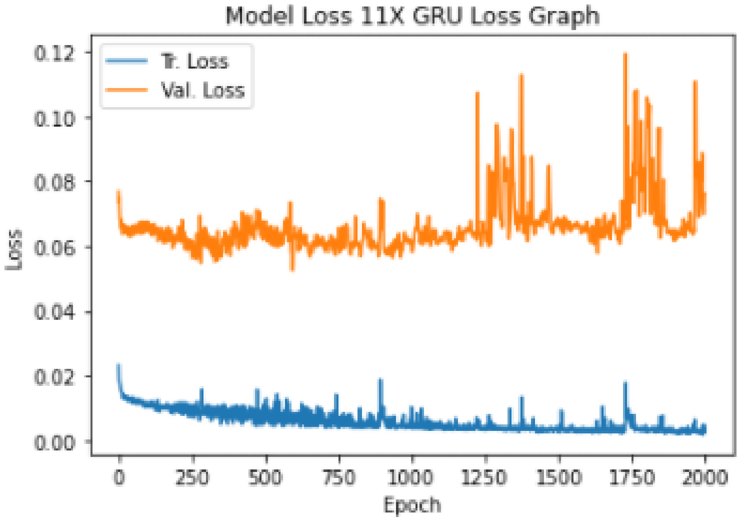


Figure 39: 11X GRU Loss Graph

B.2 32E LSTM

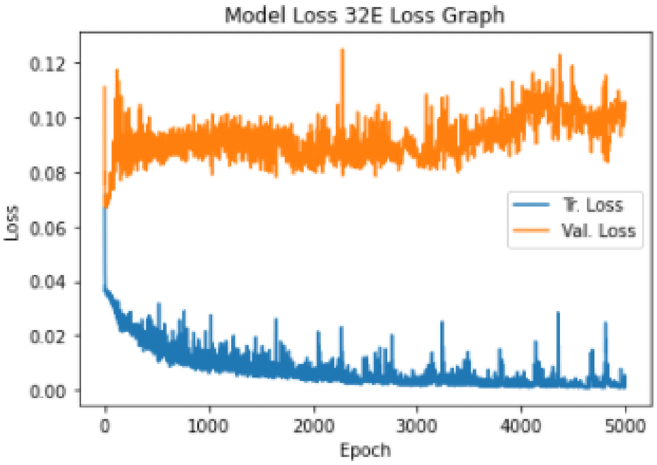


Figure 40: 32E LSTM Loss Graph

B.3 62E GRU

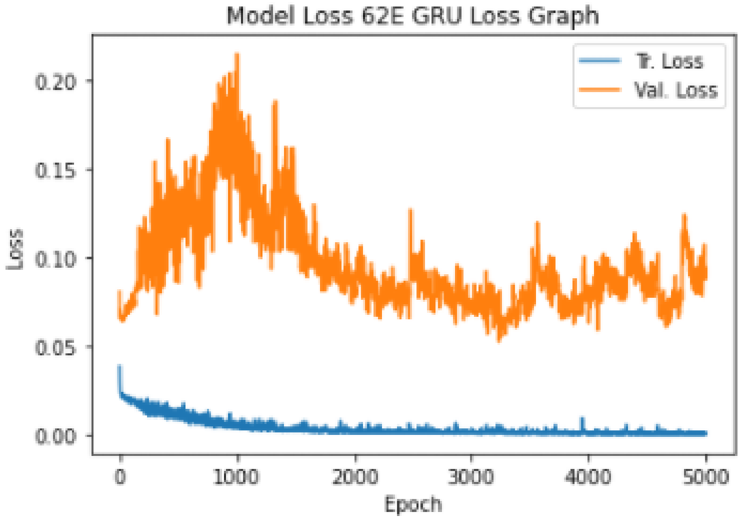


Figure 41: 62E GRU Loss Graph

B.4 63A LSTM

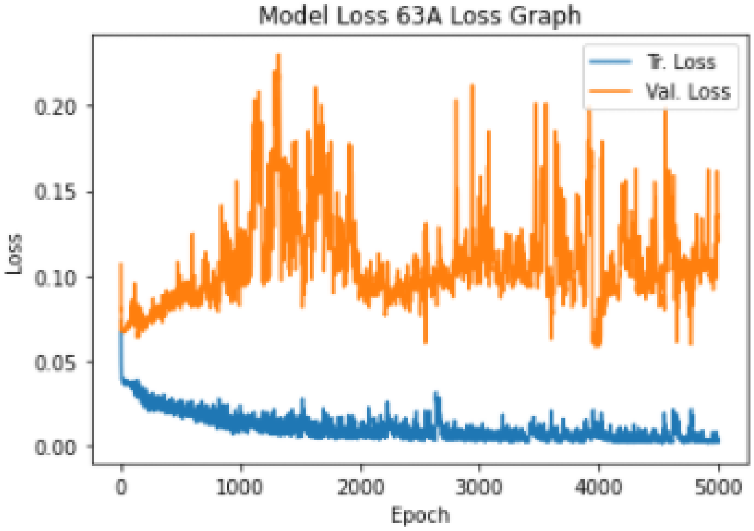


Figure 42: 63A LSTM Loss Graph

B.5 64P GRU

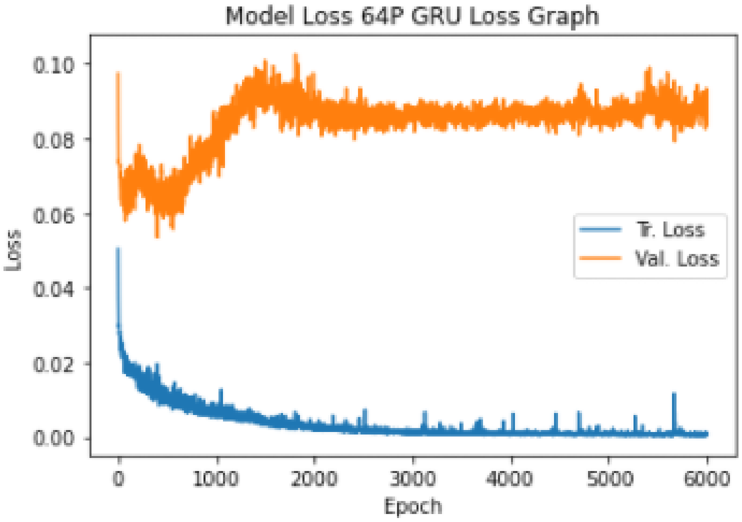


Figure 43: 64P GRU Loss Graph

Appendix C. Code Examples

C.1 LSTM Code

This is an example of the data preparation, building of a sequential model, and plotting the model prediction for the 11X LSTM model.

```
from sklearn import preprocessing as prep
from sklearn.preprocessing import MinMaxScaler
import time
import math
import tensorflow
import keras
import tensorflow.keras
from tensorflow.keras.layers import LSTM, GRU
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Activation,
   Dropout
from tensorflow.keras import callbacks
from tensorflow.keras import optimizers
from keras_tuner import *
from pickle import dump,load

def standard_scaler(X_train, X_test):
    train_samples, train_nx, train_ny = X_train.shape
    test_samples, test_nx, test_ny = X_test.shape
    X_train = X_train.reshape((train_samples, train_nx *
       train_ny))
    X_test = X_test.reshape((test_samples, test_nx *
       test_ny))

    #preprocessor = prep.StandardScaler().fit(X_train)
    scaler = MinMaxScaler(feature_range=(0, 1))

    X_train = scaler.fit_transform(X_train)
    X_test = scaler.fit_transform(X_test)

    X_train = X_train.reshape((train_samples, train_nx,
       train_ny))
    X_test = X_test.reshape((test_samples, test_nx,
       test_ny))

    return X_train, X_test

def preprocess_data(data, seq_len):
    amount_of_features = len(data.columns)
    data = data.values
    data = data.astype(’float32’)

    sequence_length = seq_len + 1
    result = []
    for index in range(len(data) - sequence_length):
```

```
    result.append(data[index : index + sequence_length])

result = np.array(result)
train_row = round(0.8 * result.shape[0])
train = result[: int(train_row), :]

result = np.array(result)
valid_row = round(0.9 * result.shape[0])
valid = result[int(train_row):int(valid_row), :]

train, result = standard_scaler(train, result)
valid, result = standard_scaler(valid, result)

X_train = train[:, : -1]
y_train = train[:, -1][: ,-1]

X_valid = valid[:, : -1]
y_valid = valid[:, -1][: ,-1]

X_test = result[int(valid_row) :, : -1]
y_test = result[int(valid_row) :, -1][ : ,-1]

X_train = np.reshape(X_train, (X_train.shape[0],
   X_train.shape[1], amount_of_features))
```

```
    X_valid = np.reshape(X_valid, (X_valid.shape[0],
       X_valid.shape[1], amount_of_features))
    X_test = np.reshape(X_test, (X_test.shape[0], X_test.
       shape[1], amount_of_features))

    return [X_train, y_train, X_valid, X_test, y_valid ,
       y_test]

def graphHistory(history, title):
    ’’’
    Function for graphing the training and validation loss
    ’’’
    # summarize history for loss
    fig = plt.figure()
    plt.plot(history.history[’loss’])
    plt.plot(history.history[’val_loss’])
    plt.title(’Model Loss ’ + title)
    plt.ylabel(’Loss’)
    plt.xlabel(’Epoch’)
    plt.legend([’Tr. Loss’, ’Val. Loss’])
    plt.show()

window = 10
X_train, y_train, X_valid ,X_test, y_valid ,y_test =
   preprocess_data(_11x[:: -1], window)
```

```
def build_best_model():
    model = Sequential()

    model.add(LSTM(input_dim = X_train.shape[2],
                   return_sequences = True,
                   units = window))
    model.add(Dropout(0.5))

    model.add(LSTM(units = 64,
                   return_sequences = True))
    model.add(Dropout(0.5))

    model.add(LSTM(units = 96,
                   return_sequences = True))

    model.add(LSTM(units = 64,
                   return_sequences = True))
    model.add(Dropout(0.5))

    model.add(LSTM(units = 64,
                   return_sequences = True))
    model.add(Dropout(0.5))

    model.add(LSTM(units = 124,
                   return_sequences = False))
    model.add(Dense(y_train.shape[0],
                   activation='relu'))

    model.compile(loss = 'mean_squared_error',
                  optimizer = optimizers.Adam(
                     learning_rate = 0.001))
    return model

model_11x = build_best_model()

history = model_11x.fit(X_train, y_train, epochs=1200,
   validation_data=(X_valid, y_valid), verbose=1, shuffle=
   False)

graphHistory(history, '11X Loss Graph')

pred, actual = inverse_scale(X_test, _11x, model_11x)

plt.plot(pred, color='red', label='Prediction')
plt.plot(actual, color='blue', label='Ground Truth')
plt.xlabel('Months')
plt.ylabel('Separation Rate')
plt.title('LSTM Test Data Prediction for 11X')
plt.legend(['pred', 'actual'])
rmse = math.sqrt(mean_squared_error(actual, pred))
print("Test RMSE:", rmse)
```

C.2 Auto ARIMA

Auto_ARIMA code example where the input is a dataframe of a specific AFSC's separation rate. The first conditional statement forces the model to take seasonality and stationarity into account. The second model is to return the best model on the basis of RMSE.

```
import numpy as np
import pandas as pd
from sklearn.metrics import mean_squared_error
import pmdarima as pm
import statsmodels.api as sm
import warnings


def autoarima(df):
    warnings.filterwarnings("ignore")
    X = df.values
    train, test = X[0:128], X[129:143]

    if adfuller(df)[1]<0.05:
        model1 = pm.auto_arima(train, seasonal=True, m=12)
        forecast1 = model1.predict(test.shape[0])
        rmse1 = sqrt(mean_squared_error(test, forecast1))
        model2 = pm.auto_arima(train, seasonal=False, m
           =12)
        forecast2 = model2.predict(test.shape[0])
        rmse2 = sqrt(mean_squared_error(test, forecast2))
    else:
```

```
    model1 = pm.auto_arima(train, seasonal=True, d =
       1, max_d=3 ,m=12)
    forecast1 = model1.predict(test.shape[0])
    rmse1 = sqrt(mean_squared_error(test, forecast1))
    model2 = pm.auto_arima(train, seasonal=False, d =
       1, max_d=3, m=12)
    forecast2 = model2.predict(test.shape[0])
    rmse2 = sqrt(mean_squared_error(test, forecast2))

if rmse1 < rmse2:
    figure = plt.figure()
    plt.plot(forecast1, color = ’red’)
    plt.plot(test, color = ’blue’)
    plt.legend([’pred’,’test’])
    print(’Test RMSE: %.5f’ % rmse1)
    print(model1)
else:
    figure = plt.figure()
    plt.plot(forecast2, color = ’red’)
    plt.plot(test, color = ’blue’)
    plt.legend([’pred’,’test’])
    print(’Test RMSE: %.5f’ % rmse2)
    print(model2)
```

Bibliography

1. Air Force Personnel Center Public Affairs JBSA. Air Force Offers Limited Active Duty Service Commitment Waivers, Expands PALACE CHASE. https://www.jbsa.mil/News/News/Article/2475674/air-force-offers-limited-active-duty-service-commitment-waivers-expands-palace/, 2021.

2. National Bureau of Economic Research. Determination of the February 2020 Peak in US Economic Activity. https://www.nber.org/news/business-cycle-dating-committee-announcement-june-8-2020, 2020.

3. L.R. Klein and S. Ozmucur. Some Possibilities for Indicator Analysis in Economic Forecasting. Pami Dua (ed.) Business Cycles and Economic Growth: An Analysis Using Leading Indicators. *Oxford University Press*, pages 243–257, 2004.

4. M.K. McGee. Modeling Air Force Retention with Macroeconomic Indicators. Master's thesis, Air Force Institute of Technology, 2015.

5. G.J. Whelan. Forecasting Army Enlisted ETS Losses. Master's thesis, Naval Postgraduate School, 2013.

6. G. Panchal, A. Ganatra, Y.P. Kosta, and D. Panchal. Forecasting employee retention probability using back propagation neural network algorithm. *IEEEXPLORE*, May 2010.

7. N. Ben Yahia, J. Hlel, and R. Colomo-Palacios. From big data to deep data to support people analytics for employee attrition prediction. *IEEE Xplore Full-text PDF:*, Apr 2021.

8. G.P. Martin, R.R. Hill, and G. Bailey. An Artificial Life Approach to Managing Pilot Retention. 1999.

9. H.L. Jantscher. An Examination of Economic Metrics as Indicators of Air Force Retention. Master's thesis, Air Force Institute of Technology, 2016.

10. J. Elliot. Air Force Officer Attrition: An Econometric Analysis. Master's thesis, Air Force Institute of Technology, 2018.

11. J.A. Schofield. Non-Rated Air Force Line Officer Attrition Rates Using Survival Analysis. Master's thesis, Air Force Institute of Technology, 2015.

12. T.S. Pujats. Forecasting Attrition by AFSC for the United States Air Force. Master's thesis, Air Force Institute of Technology, 2020.

13. Z. Tang, C. De Almeida, and P.A. Fishwick. Time series Forecasting Using Neural Networks vs. Box-Jenkins methodology. *Simulation*, 57(5):303–310, 1991.

14. B. Kordanuli, L. Barjaktarović, L. Jeremić, and M. Alizamir. Appraisal of Artificial Neural Network for Forecasting of Economic Parameters, 2017.

15. P.G. Zhang. Time series Forecasting Using a Hybrid ARIMA and Neural Network Model. *Neurocomputing*, 50:159–175, 2003.

16. A. Géron. *Hands-on Machine Learning with Scikit-Learn and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems.* O'Reilly, second edition edition, 2019.

17. E. Tölö. Predicting Systemic Financial Crises with Recurrent Neural Networks. *Financial Stability*, 49, 2020.

18. S. Siami-Namini and A. Siami-Namini. Forecasting economics and financial time series: Arima vs. lstm. `https://arxiv.org/abs/1803.06386`, 2018.

19. L. Di Persio and O. Honchar. Recurrent neural networks approach to the financial forecast of google assets. *International journal of Mathematics and Computers in simulation*, 11:7–13, 2017.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From — To)*
24–03–2022	Master's Thesis	March 2021 — Mar 2022

4. TITLE AND SUBTITLE

An Exploratory Analysis of Time Series Econometric Data for Retention Forecasting Using Deep Learning

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

O'Donnell, John, 2nd Lt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENS-MS-22-M-159

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Headquarters Air Force (HAF/A1XD)
Douglas A. Boerman
1550 W. Perimeter Rd., Rm 4710
Joint Base Andrews NAF Washington, MD 20762-5000
Email: douglas.a.boerman.civ@mail.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

HAF/A1XD

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Officer retention in the Air Force has been researched many times in an attempt to better predict the personnel needs of the Air Force for the future. There has been previous work done in regards to specific AFSCs and how their retention compares to specific yet similar private sector jobs. This study considers different econometric time series statistics as a feature space and an average Air Force officer separation rate as the response variable for the multivariate time series analysis deep learning techniques. The econometric indicators used in this study are New Business Formations, New Durable Good Orders, and the Consumer Confidence Index. The techniques considered for this study were Long Term Short Memory(LSTMs) Networks and Gated Recurrent Unit(GRU) Networks. This study shows that both GRUs and LSTMs perform fairly well with a forecast of 14 months out, but does not perform well comparatively to the more traditional univariate time series forecasting techniques, ARIMA models. The career fields with better performing models were career fields that will have jobs outside of the Air Force that will be more likely to hire in a period of economic growth, which would in turn increase the separation rate.

15. SUBJECT TERMS

artificial neural network, ANN, deep learning, machine learning, time series, recurrent neural networks, RNN, long short-term memory networks, LSTMs, gated recurrent units, GRUs, forecasting, personnel models, econometric models

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. Raymond Hill, AFIT/ENS
					19b. TELEPHONE NUMBER *(include area code)*
U	U	U	UU	73	(937) 255-3636 x7469; rhill@afit.edu